REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	12/6/96	Final Report, 12/15/93 - 10/31/96

4. TITLE AND SUBTITLE

Communication and Coordination in Multi-Agent Systems: Agent-Oriented Programming and Computational Social Laws.

5. FUNDING NUMBERS

F49620-94-1-0090

6. AUTHOR(S)

Yoav Shoham

AFOSR-TR-97

0359

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Dept. of Computer Science
Stanford University
Stanford, CA 94305

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NM
Department of the Air Force
110 Duncan Avenue, Suite B115
Bolling AFB, DC 20332-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release: distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Agent oriented programming was proposed as a high-level programming language, in which a programmer is given the opportunity to communicate with other programs in a uniform, high-level language. Furthermore, the programmercould explicitly represent in the program (or 'agent') the relationship with other program (or 'agent'), including the beliefs about the other agents and the obligationsmade to them. Our hypothesis was that such 'mental-level' design would provide a powerful abstraction that would enable the analysis and even design of complex distributed systems.

In addition to such coordination via high-level modeling and communication, we were interested in global mechanisms that eliminate the need for explicit coordination in the place. Specifically, we borrowed from everyday experience the notion of social laws and conventions. The idea is that just as in real life traffic rules restrict one enough to eliminate most the need for real-time conflict resolution while driving but not so much so as to make any navigational goal unattainable, so could restrictions on computation strike a good balance. These restrictions could either be imposed directly by a system-or network-administrator, or could emerge dynamically through a process of trial an error in the population

14. SUBJECT TERMS

agent-oriented programming, mental state, load balancing, coordination

15. NUMBER OF PAGES

3

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	unlimited

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only (*Leave blank*).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C -	Contract	PR -	Project
G -	Grant	TA -	Task
PE -	Program Element	WU -	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- DOE - See authorities.
- NASA - See Handbook NHB 2200.2.
- NTIS - Leave blank.

Block 12b. Distribution Code.

- DOD - Leave blank.
- DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- NASA - Leave blank.
- NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases entifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Final Report on AFOSR grant AF F49620-94-1-0090:
Communication and coordination in multi-agent systems:
agent-oriented programming and computational social laws.

Yoav Shoham
Director, Robotics Laboratory
Computer Science Department
Stanford University

1 Background

The premise our research has been that in emerging networked environments it will increasingly be the case that neither authority nor information are concentrated in a single locus, and software will have to be written in a way that reflects this fact. Specifically, we posited an environment in which multiple programs operate, each controlled by and embodying the wishes of different masters. These programs would require to coordinate with one another, whether to achieve tasks neither can achieve alone or to resolve conflicts around shared resources (including computational resources such as network printers and other resources such as transportation vehicles). We set out to investigate two new types of mechanism aimed at achieving such coordination.

Agent oriented programming was proposed as a high-level programming language, in which a programmer is given the opportunity to communicate with other programs in a uniform, high-level language. Furthermore, the programmer could explicitly represent in the program (or 'agent') the relationship with other program (or 'agents'), including the beliefs about the other agents and the obligations made to them. Our hypthesis was that such 'mental-level' design would provide a powerful abstraction that would enable the analysis and even design of complex distributed systems.

In addition to such coordination via high-level modeling and communication, we were interested in global mechanisms that eliminate the need for explicit coordination in the first place. Specifically, we borrowed from everyday experience the notion of social laws and conventions. The idea is that just as in real life traffic rules restrict one enough to eliminate most of the need for real-time conflict resolution while driving but not so much so as to make any navigational goal unattainable, so could restrictions on computation strike a good balance. These restrictions could either be impose directly by a system- or network-administrator, or could emerge dynamically

through a process of trial an error in the population.

Below is a summary of our achievements on these efforts during the period of the contract. Following it are a few representative publications.

2 AOP and mental-level modeling

Our first experience was, in a sense, negative. The framework of AOP, while very attractive conceptually, proved too high-level to be useful in particular applications. While we experimented with several, including distributed transportation planning and network management, the details of the particular application ended up dominating the power of AOP itself. Our tentative conclusion from this experience that AOP is useful as a design principle, but each domain calls for specific AOP-inspired language. We are currently investigating, not under the AFOSR contract, applying these lessons to inter-application communication standards.

At the same, our experience in applying the mental state component of AOP alone met with success. Up until the time of this research, logics of knowledge (that is, logics in which one can state what is known by a oarticular agent, not only what is true or false) were applied solely to reason about distributed protocols. We were able to show that these same tools, albeit in modified form, are useful in the robotic domain. Specfically, we were able to synthesize provably optimal termination conditions for robot motion planning, and were furthermore able to present an algorithm to automatically distribute a centralized robotic controller among the different robotic components. These results are reported in two of the attached papers.

3 Computational social systems

Our results here are quite crisp, and are presented four attached papers. In the first paper we investigate the difficulty of sythesizing useful social laws by carefully analysing the domain at hand. We first perform a case study of manual construction of traffic laws in a grid system, and then investigate the computational complexity of automatically synthesizing such social laws.

In a second paper we investigate the automatic emergence of such laws (and specifically, social conventions), as a result of a stochastic process in a population. This investigation is carried out in a mathematical setting, and so in a third paper we investigate the pheneomena in a quasi-realistic load-balancing setting. Specifically, we create a setting in which n processes

stochastically generate jobs, each of which must be submitted to one of m processors, where each processor has some varying and unknown capacity. We show how the processes learn to distribute the jobs optimally without any global information or other hints, based purely on their accumulated individual histories.

Finally, we apply these ideas in a truly real-world setting. In the fourth paper attached herein we report on results with an adaptive information retrieval system. The goal of this system, (called Fab) is to fetch users Web pages and over time, based on feedback, home in on the users' interests. Fab employs both content-based and collaborative components. Importantly, at the core of Fab's architecture are a set of collection agents, which must between them perform optimal search of a very large space (the Web) in service of an unknown and ever changing set of users. Fab has been operational for a couple of years now; it has proved very fertile ground in which to investigate the emergence of coordination, and is also now beginning to attract much attention due to its impressive performance and friendly design.